# ZUR ANWENDUNG WISSENSBASIERTER SYSTEME IM VERKEHRSWESEN

Hochschuldozent Dr. sc. techn. MANFRED KOSEMUND, Dresden/DDR
Dipl.-Ing. KAMEN DANOWSKY, Sofia/VR Bulgarien und
ANDREAS SCHMIDT, Dresden/DDR

## 1. Neue Beschreibungs- und Modellbildungsformen

Die Rechentechnik/Datenverarbeitung ist aus dem Bedürfnis heraus entstanden, große Mengen von Zahlen zu verarbeiten. Man ist somit gewohnt, mit Zahlen zu rechnen und mit Zahlen auch zählbare und meßbare Größen, z.B. aus der Physik zu verarbeiten. Damit lassen sich gleichfalls bestimmte Modelle objektiver Realitäten, nämlich die mit zähl- und meßbaren Größen beschreibbar sind, bilden. Bemerkenswert ist jedoch, daß dabei in das Datenverarbeitungsmodell nur die Maßzahlen, nicht die Begriffe und Maßeinheiten eingehen.

Eine Erweiterung der Beschreibungsmöglichkeiten brachte die Einführung der Zeichenketten, mit der die Schriftsprache im Sinne der üblichen Textverarbeitung, aber auch syntaktische Begriffsbanken (Datenbanken) nutzbar gemacht werden konnten. Eine ähnliche Erweiterung war die Abspeicherung und Manipulation von Graphikelementen (Linien, Kreisbögen u.a.) sowie von Bildern (Punktmustern).

Allen gemeinsam ist die Manipulierbarkeit durch Befehlsfolgen (Programm) , mit denen Algorithmen beschrieben werden können. Dabei ist die leichte Trennbarkeit von Datenbestand und Datenverarbeitung durch Programme charakteristisch.

Die besonders durch "Expertensysteme" bekannt gewordene "Wissensverarbeitung" geht jedoch einen anderen Weg, der durch die Beschreibungssprache PROLOG wohl am deutlichsten ausgedrückt wird. Die sogenannte "Wissensbasis" besteht aus Fakten, die in Anlehnung an die Prädikatenlogik als Aussagen formuliert werden. Anders als bei den bekannten natürlichen Sprachen steht dabei die Aussage am Beginn eines Satzes, gefolgt von Subjekt-Objekt-Beziehungen.

Zum Beispiel heißt die Aussage
"Zug D371 ist ein Schnellzug"
in PROLOG

```
ist_ein_schnellzug (zug_d371).
```

oder auch (mit Subjekt und Objekt)

```
ist_ein (schnellzug, zug_d371).
```

Mit der Einführung von Variablen (gekennzeichnet durch Großschreibung) für die Argumente wird aus der Aussage (fact) eine Regel (rule), z.B.

```
ist_ein (Zuggattung, Zug).
```

Sie hat die allgemeingültige Aussage, daß durch "ist_ein" das Subjekt "Zug" zur Klasse "Zuggattung" gehört.
Eine wesentliche Erweiterung des Regelinhaltes besteht darin, daß die Aussage einer Regel an Bedingungen geknüpft werden kann. Zum Beispiel ist (im Tarif der DR) ein Zug ein Schnellzug, wenn sein Gattungszeichen (Variable GATTZ) die Werte d (= Schnellzug) oder iex (= Interexpreßzug) annimmt. Die damit zu formulierenden Regeln und Fakten (= Wissensbasis) lauten:

```
ist_ein (schnellzug, Zug)                (1)
      :-gehört_zu (d, GATTZ)
      ; gehört_zu (iex, GATTZ)
gehört_zu (GATTZ, Zug).                  (2)
gehört_zu (iex, iex73).                  (3)
```

Die Zeichen ":-" bedeuten "if (wenn-dann) bzw. ";" die Alternative "oder". Üblicherweise wird "oder" vermieden und dafür zwei Regeln geschrieben.

An diese Wissensbasis kann z.B. die Frage gestellt werden, ob der Zug iex73 ein Schnellzug ist. PROLOG findet in der Regel (1), daß ein Zug ein Schnellzug sein kann, wenn die nachstehenden Bedingungen erfüllt sind. Es weist der Variablen Zug den Wert iex73 zu und stellt fest, daß die Regel erfüllbar ist, wenn die Variable GATTZ mit einem geeigneten Wert belegt werden kann. In der Regel (2) wird

festgestellt, daß GATTZ mit der schon mit dem Wert iex73 belegten Variablen Zug zusammenhängt. Das Faktum (3) schließlich zeigt, daß zu iex73 die Gattung iex gehört und somit eine der alternativen Bedingungen in (1) erfüllbar ist, somit auch der Kopf der Regel (1) wahr ist.

Zur Beantwortung der diskutierten Frage benötigt PROLOG (im Gegensatz zu algorithmischen Sprachen) kein Programm. Der zur Beantwortung der Frage notwendige Algorithmus ist in PROLOG impliziert und besteht grundlegend darin, Variablen Werte der Fragestellung zuzuweisen (matching), um (im allgemeinen Fall durch Einsetzung in weitere Regeln) eine "wahre" Aussage zu finden.

Das die 2. Bedingung erfüllbar ist, wird mit Hilfe des Faktes, daß der Zug iex73 die Gattung iex besitzt, geprüft. Natürlich wird in gleicher Weise die erste Bedingung der zweiten Regel geprüft.

Wenn mit einer gewählten Wertzuweisung eine "wahre" Aussage nicht realisierbar ist, wird die Bindung wieder aufgelöst (backtracking) und eine neue weiterverfolgt, solange, bis eine Lösung gefunden ist, oder es sicher keine Lösung gibt.

Der wesentlichste Vorteil besteht darin, daß die Wissensbasis beliebig erweiterungsfähig ist, ohne daß dabei neue Algorithmen zu formulieren sind (da überhaupt keine formuliert werden) oder vorgegebene Strukturen (wie bei Datenbanken) einzuhalten sind. Schwierig ist bei größeren Wissensbasen, daß (wie auch bei algorithmischen Sprachen) der semantische Inhalt (der syntaktisch durch die Position in einer Regel bzw. durch Zeichenketten, dagegen nicht eindeutig durch Variablennamen festgelegt wird) nicht implizit bestimmt ist, sondern in geeigneter Weise getrennt dokumentiert werden muß.

Ähnlich wie bei einer im Gehirn des Menschen gespeicherten Wissensbasis werden natürlich bei falschen Regeln auch falsche Schlüsse gezogen. Diese werden beim Menschen in den meisten Fällen durch nahezu unbewußte Plausibilitätstests als falsch erkannt. Bei einer rechnergestützten Wissensbasis funktioniert dieser Mechanismus nur, wenn er tatsächlich zur Wissensbasis gehört. Da die Nutzung des Wissens wesentlich freizügiger als bei einer algorithmischen Sprache ist, sind die Fehlermöglichkeiten entsprechend größer. Dieser Asist demzufolge sorgfältig zu beachten.

## 2. Konzept für ein wissensbasiertes Zugauskunftsaystem

Als ein erster Test der Anwendungsmöglichkeiten, aber auch zum Erlernen des Umgangs mit einer Wissensbasis anstelle von Dateien und Programmen wurde ein Zugauskunftsaystem entwickelt [2]. Bestandteil des Systems ist eine Datenbasis als Auszug aus einem Eisenbahn-Kursbuch. Diese Datenbasis (Fakten) besteht aus folgenden Informationen:

(1) einem Verzeichnis von Strecken (als Äquivalent des Streckenverzeichnisses bzw. einer Streckenkarte)

(2) einem Verzeichnis der Fahrzeiten der über die Strecken verkehrenden Züge

Neuartig ist die Formulierung von Regeln zur Suchstrategie in einer kursbuchanalogen Datenbasis. Die prinzipielle Lösung wird nachfolgend grob skizziert. Ein Ausschnitt aus der Datenbasis nach (1) lautet beispielsweise:

(1.1) verbindung (dresden, leipzig, riesa).
(1.2) verbindung (dresden, leipzig, döbeln).
(1.3) verbindung (halle, erfurt, weissenfels).
(1.4) verbindung (leipzig, erfurt, weissenfels).

wobei die Interpretation des ersten Faktums lautet:

"Es existiert eine Verbindung von Dresden nach Leipzig über Riesa". Maßgeblich für die Art der (eingeklammerten) Eigenschaften der Aussage (von, nach, über) ist allein die Position innerhalb der Klammer.

Ein ernsthaftes Verständnisproblem ist dabei durch die Tatsache gegeben, daß die verbal spezifizierte Aussage (es existiert, von, nach, über) zwar in dem gewollten Sinn (bei richtiger Formulierung der Regeln) berücksichtigt wird, aber explizit nicht erscheint.

Zur Nutzung der formulierten Verbindungen lassen sich folgende Regeln formulieren:

(3) reise (X,Y,Z):- verbindung (X,Y,Z)

mit der Interpretation: "Man kann eine Reise von X nach Y über Z durchführen, wenn eine Verbindung von X nach Y über Z existiert". Mit Hilfe dieser Regel wären z.B. folgende Fragen beantwortbar

- Kann man von Dresden nach Leipzig reisen?
- Kann man nach Erfurt reisen?
- Welche Reisen nach Leipzig führen über Döbeln?

Die erste der Fragen z.B. würde, in PROLOG formuliert, lauten:

(3) ? reise (dresden, leipzig, _ )

Sie würde damit beantwortet, daß entsprechend der Klausel (1.1) eine Verbindung von Dresden nach Leipzig über Riesa existiert und demzufolge über diese Relation gereist werden kann. (Wollte man alle Verbindungen zwischen Dresden und Leipzig erfassen, müßte man die Frage anders formulieren.)

Die Fragestellung, ob man auch von Dresden nach Erfurt reisen kann, würde, entgegen den praktischen Erfahrungen mit Benutzung der Regel (3) verneint. Das heißt, das Wissen der Basis ist unzureichend. Es berücksichtigt nicht die Tatsache, daß ein Ziel auch durch ein- oder mehrmaliges Umsteigen zu erreichen ist. Für ein einmaliges Umsteigen kann die Regel in der Form

```
reise (X,Y, (Z1, Z2)):-
            verbindung (X,U,Z1),
            verbindung (U,Y,Z2).
```

formuliert werden, die besagt, daß eine Reise von X nach Y auch dann möglich ist, wenn ein Unterwegsbahnhof U vorhanden ist, der die Funktion des Zielbahnhofs in einer Relation und die des Startbahnhofs in einer zweiten Relation erfüllt (im Beipiel (1.1) und (1.4). Diese Regel kann auch rekursiv formuliert werden, um ein Ziel durch mehrmaliges Umsteigen zu erreichen. In ähnlicher Form, wie hier für die Verbindungssuche dargestellt, ist auch die Suche nach Zügen, die über die gefundenen Relationen verkehren, möglich, ohne daß der Vorgang hier dargestellt wird. Ein wesentlicher Vorteil der Nutzung eines wissensbasierten Systems, ist der, daß mögliche Fragestellungen nicht a priori in die Wissensbasis eingebracht werden müssen. Dieser wesentliche Vorteil gegenüber anderen Programmiersprachen wird allerdings dadurch aufgehoben, daß diese Fragen (goals) natürlich in PROLOG mit den in der Wissensbasis benutzten Begriffen formuliert werden müssen. Für den praktischen Betrieb ist das selbstverständlich unzumutbar. Der Ausweg, der benutzt wurde, besteht in einer Menüführung, die aber die universell mögliche Fragestellung auf eine durch das Menü vorgegebene reduziert (und damit wieder algorithmiert). Ein letztlich unumgänglicher anderer, aber komplizierterer Ausweg besteht in der Formulierung von Fragen in natürlicher Sprache. Beispielsweise wäre eine Formulierung der Frage (3) in der Form "Besteht die Möglichkeit, von Dresden nach Leipzig zu fahren?" denkbar. Das System müßte daraus ableiten, daß das Verb "fahren" auf das Prädikat "reise" zielt, daß "von" das Wort "Dresden" dem Startbahnhof zuordnet, daß "besteht die Möglichkeit" eine Umschreibung für eine Fragestellung ist, usw. Diese

- Analyse einer natürlichsprachlichen Frage,
- die Antwortfindung durch logische Analyse der Wissensbasis,
- die natürlich formulierte Ausgabe der Antwort und
- erforderlichenfalls eine Erklärung der Schlußkette

bilden die wesentlichen Komponenten eines Expertensystems.

Ein Vergleich des deklarativ (in PROLOG) formulierten Zugauskunftssystems mit einem algorithmisch programmierten ergibt Aufwendungen in vergleichbaren Größenordnungen, wobei unter dem Gesichtspunkt der durch die auch bei PROLOG notwenige Einschränkung durch menügeführte Fragestellungen der algorithmische Weg sogar Vorteile aufweist.

In der Literatur (z.B. in [1]) wird angegeben, daß entsprechend Bild 1 bei zunehmender Komplexität der Probleme bei algorithmischer Programmierung schließlich eine nicht überwindbare "Schallmauer" des Aufwandes erreicht wird, die nur durch deklarative Programmierung durchbrochen werden kann. Nach Ansicht des Autors ist beim Vergleich der algorithmischen mit den deklarativen Beschreibungsformen nicht so sehr die sog. "Schallmauer" interessant (von der

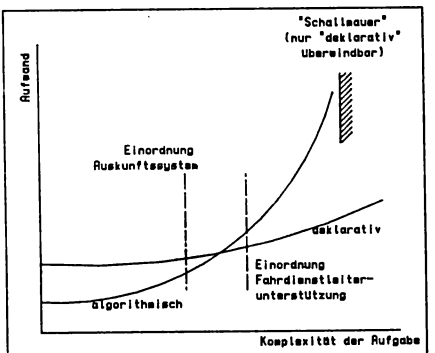


Bild 1: Qualitativer Vergleich des Aufwandes bei algorithmischer und deklarativer Programmierung (nach [1])

nicht sicher ist, ob sie überhaupt existiert), sondern der Schnittpunkt der Linien, die die Aufwendungen verkörpern, d.h. die Trennstelle der Bereiche, bei denen die eine oder andere Programmierungsart zweckmäßiger ist.

## 3. Nutzung einer Wissensbasis für Aufgaben des Zugfahrdienstes einer Eisenbahn

Nicht wenige Unregelmäßigkeiten im Zugbetrieb und sogar Bahnbetriebsunfälle haben ihre Ursache darin, daß die Beschäftigten in Ausnahmesituationen den Teil ihrer Vorschriften, den sie, wegen der Seltenheit dieser Situationen, nicht routinemäßig nutzen, nicht immer im nötigen Umfang beherrschen. In dieser Situation einen Experten zu befragen, ist oft aus zeitlichen und örtlichen Gründen nicht möglich. Oft werden auch vorhandene Möglichkeiten aus psychologischen Gründen (Eingeständnis von Unwissen oder Unsicherheit) nicht genutzt. Für das nochmalige Studium der einschlägigen Vorschriften reicht in der Regel die verfügbare Zeit nicht aus.

Bei der Installation rechnergeführter Betriebssysteme besteht die Zielstellung in der Beherrschung des regulären Betriebes mit einem Minimum an personellem Aufwand. Bezieht man viele Ausnahmesituationen in die Programmsysteme ein, besonders durch den Ausfall beliebiger technischer Komponenten hervorgerufen, werden sehr schnell die Grenzen der rechentechnischen Möglichkeiten erreicht. Für den verbleibenden Rest müssen dann weiterhin Entscheidungen zur Betriebsführung manuell getroffen werden.

Daraus wird die Notwendigkeit begründet, nach neuen Hilfsmitteln, die in diesen Situationen wirksam werden, zu suchen. Eine neue und interessante Möglichkeit bilden wissensbasierte Systeme. Unter den Voraussetzungen, daß sie dort genutzt werden, wo auch der reguläre Prozeß rechnergestützt betrieben wird, ist die Implementation leicht möglich. Es ist außerdem nicht schwierig, den potentiellen Nutzer zu zwingen, die Wissensbasis im Bedarfsfall auch zu nutzen.

Zur Realisierung der Zielstellung muß die Wissensbasis, analog dem Wissen beispielsweise eines Fahrdienstleiters, folgende Komponenten enthalten:

(1) die Regeln der Fahrdienstvorschrift und anderer einschlägiger Vorschriften

(2) Fakten und Regeln der Topologie des Einsatzbereiches

(3) Fakten und Regeln der örtlich und zeitlich geltenden Pläne zur Durchführung der Betriebsaufgabe (aktueller Fahrplan u.a.)

(4) den aktuellen Prozeßzustand (Stellung der Weichen und Signale, Gleisbelegungen) und

(5) die prinzipielle Funktionsweise einer Eisenbahn.

Mit dem letztgenannten Punkt wird der Tatsache Rechnung getragen, daß bei der menschlichen Wissensverarbeitung ständig eine Fülle von Hintergrundwissen nahezu unbemerkt mit verarbeitet wird. Erfahrungsgemäß machen sich in einer rechnergestützten Wissensbasis hier am ehesten Wissenslücken bemerkbar. (Das System hat kein Allgemeinwissen!).

Als Test, wie die angegebene Realisierung eines wissensbasierten Systems zur Unterstützung der Arbeit eines Fahrdienstleiters ermöglicht werden kann, wurde der Arbeitskomplex "Sperrfahrten" ausgewählt, d.h. die Fakten und Regeln programmiert, die beim Befahren eines eigentlich gesperrten Streckenabschnittes zu beachten sind [3]. Die Fahrdienstvorschrift gibt hierzu einen Algorithmus für die beteiligten Eisenbahner vor und nennt die Bedingungen, unter denen der Algorithmus abgearbeitet werden darf. Ein algorithmischer Teil läßt sich in eine deklarative Beschreibung leicht einbeziehen. Grob vereinfacht dienen dazu Regeln in der Form

sperrfahrt :- anzeige (liste anzeigeelemente),
eingabe (werte).

Mit der Eingabe "? sperrfahrt" als "Was soll ich bei einer Sperrfahrt tun?" wird die entsprechende Regel gesucht. Diese ist an die Bedingungen geknüpft, daß zuerst eine Anzeige auf dem Bildschirm erfolgen muß. Diese Bedingung ist immer erfüllbar und ist erfüllt, wenn die Anzeige an das Bildschirmgerät ausgegeben ist. Gleiches gilt, wenn die erforderlichen Werte eingegeben sind. Die Verarbeitung erfolgt dann nach ähnlichen Gesichtspunkten, wie am Beispiel der Zugauskunft beschrieben wurde.

Bei der Beschreibung der Topologie eines Bahnhofs und einer Strecke besteht ein wichtiger Gesichtspunkt darin, daß das Wissen dieses Teils der Wissensbasis nicht für eine spezielle Aufgabe sondern universell benutzbar sein soll. Hierbei bestehen enge Beziehungen zur Wissensdarstellung (Faktenrepräsentation) einer Datenbank. Die Unterschiede der Auflistung z.B. von Gleisabschnitten und ihrer Eigenschaften in einer REDABAS/dBASE-Daten-

bank und einer PROLOG-Liste sind rein formaler Natur. Grundsätzliche Unterschiede bestehen dagegen in

- der allgemeingültigen Beschreibung von Zusammenhängen (z.B. von Gleisabschnitten) durch Regeln in PROLOG und
- der speziellen Zuordnung von Feldinhalten (die z.B. Gleisabschnitte beschreiben) in einer algorithmischen Sprache.

Ein Kernproblem (und das ist wieder unabhängig von algorithmischer oder deklarativer Beschreibung) ist die Modellbildung eines solchen Systems, daß mit rechentechnischen Mitteln bearbeitet werden soll. Die wichtigste Forderung besteht darin, daß sich für möglichst viele Aufgaben eine geeignete Untermenge von Informationen aus der Datenbasis abheben läßt. Gleichzeitig soll die Datenbasis möglichst beschränkt sein, da der Aufwand für Erstellung und Pflege natürlich proportional der Größe der Datenbasis ist. Die Ergebnisse eines Programms zur Realisierung von Sperrfahrten sind insofern ermutigend, daß der grundsätzliche Nachweis der Eignung von PROLOG für eine derartige Aufgabenklasse erbracht wurde. Ohne daß ein direkter Vergleich geführt worden ist, wird eingeschätzt, daß die Beschreibung in PROLOG effektiver als in einer algorithmischen Sprache ist. Das gilt umso mehr, weil das meiste angesammelte (und strukturierte) Wissen sich nicht nur auf die Aufgabe "Sperrfahrt" bezieht, sondern allgemeingültiger ist. Die Fortführung der Arbeiten wird nun nicht nur darin bestehen, für weitere Aufgaben einfach Wissen anzulagern, sondern analog dem menschlichen Wissenserwerb durch Schluß vom speziellen zum allgemeinen Wissen spezielle Regeln durch allgemeine Regeln iterativ zu ersetzen.

Literatur

[1] Lantermann, B.: Die Rolle der Software Anfang der 90er Jahre; Vortrag auf der online '87, Hamburg 1987

[2] Danowsky, K.: Ein wissensbasiertes Auskunftssystem zur Reisendeninformation;Diplomarbeit, HfV 1988

[3] Schmidt, A.: Untersuchung zur Darstellung fahrdienstlicher Vorschriften in der Form einer rechnergestützten Wissensbasis; Großer Beleg, HfV 1989